

## 7.0 SYSTEM AND APPLICATION DESIGN

### 7.1 CDE IMPLEMENTATION IN THE DII<sup>1</sup>

#### 7.1.1 The Desktop in DII Systems

DII systems installed on UNIX-based platforms use CDE to provide desktop management functions. The desktop configuration defined in this style guide is the default to be used by these systems unless operational requirements call for an alternative configuration (see section 1.5).

##### 7.1.1.1 The CDE Front Panel

The functionality in the system is available from the Front Panel on the desktop. The default location of the Front Panel is near the bottom of the screen. The Front Panel can be moved by dragging the title bar or by using the move handles located along the sides of the panel; it can be minimized by clicking the Minimize button in the right corner of the panel.

The Front Panel provides access to various desktop managers (i.e., Application Manager, File Manager, Style Manager, Help Manager), generic utilities (e.g., Calendar, Printer, and Trash Can) and indicators (e.g., a clock), and workspace buttons for switching between available workspaces. The remainder of section 7.1 addresses the desktop management capabilities provided by CDE that are to be implemented in the system; Help Manager is addressed in section 11 on user support resources.

The system designates a default printer that is available to all applications. This printer is shown in the Front Panel, with information about other printers provided by the Print Manager application. Users can select any printer available to the system or use the default printer if they do not specify a printer name. They can submit a job for printing by using any of the Print features available on the desktop or by selecting a Print command from within the application.<sup>2</sup>

##### 7.1.1.2 DII Configuration

Modifying desktop functions. The system provides access to desktop and manager functions in a CDE-compliant manner unless the specifications presented here indicate otherwise. Individual applications determine the constraints for movement and deletion of objects created by the application, with users allowed to perform these actions if they are granted permission (i.e., access) to do so. If these actions are allowed, they are performed using the functionality provided by the desktop. If these actions are not allowed or do not apply, the corresponding features in the desktop are unavailable (i.e., are grayed out) when the application or file icons are selected; applications are not to alter how these functions are performed or remove them from the desktop.

Classification bar. The system provides a classification bar with markings indicating the current classification level.<sup>3</sup> The bar extends across the top of the screen and cannot be obscured by application windows when they are displayed or repositioned on the screen. The classification level appears in the middle of the bar; if desired, various status indicators (such as alerts) may be indicated at the left

---

<sup>1</sup> A default desktop implementation for Windows-based systems will be addressed in a future version of this document.

<sup>2</sup> Other desktop utilities will be addressed in a future version of this document.

<sup>3</sup> Compliance with the style guidelines for the Compartmented Mode Workstation, which is included as an appendix to the DoD style guide, will be addressed in a future version of this document.

margin, and a digital date-time clock may be displayed at the right margin of the bar. Appendix D lists the colors to be used in the classification bar. All classification terms are presented in upper-case letters and spelled out, with caveats abbreviated in accordance with relevant security manuals and directives and with no embedded spaces within words in the label (e.g., SECRET, not S E C R E T).

### **7.1.2 Session Management**

#### **7.1.2.1 Starting and Ending a Session**

All users are required to enter a valid identification and password before a desktop session is initiated. All users must successfully complete this login procedure before they can access any system functions.

Users can lock a system (e.g., to prevent unauthorized use when they are not present) by clicking on the Lock control in the Front Panel and to unlock the system by typing their password. In addition, users can configure the screen saver and lock features for the system using controls in Style Manager.

To end a session, users click on the Exit control in the Front Panel or choose the Log Out option in the Workspace menu. Based on their Logout preference setting in Style Manager, users may be prompted to confirm this action. When users log out of the desktop, the current session is saved so that when they log in again, the desktop appears as it did at the end of the previous session. When a session ends, all application windows are closed, and when logout is complete, a login window is displayed on the screen.

#### **7.1.2.2 DII Configuration**

Login. The system does not allow users to access a Terminal window unless this privilege has been granted to them. In addition, the Failsafe Session and Command Line Login features in CDE are removed. Finally, the system does not provide options to select a language other than English unless other languages are supported (see section 12 on internationalization).

Session. CDE provides two options with regard to session configuration. In the “current session” option, which is the default, the desktop saves session information (e.g., current appearance and behavior settings, applications that are running) when the user logs out and restores the system using this information when the user logs in again. In the “home session” option, users can explicitly save a particular session configuration and then choose to have that session restored whenever they log in. The system supports both of these options and allow users to select between them in Style Manager.

### **7.1.3 Application Management**

#### **7.1.3.1 Accessing Applications and Application Groups**

The system uses Application Manager to provide access to the collection of applications available on the system. The applications can be represented as individual icons or arranged in groups which are represented by a container icon. Four application groups containing various tools and utilities are built into the desktop. Desktop\_Apps includes desktop applications such as File Manager, Style Manager, and Calculator. Desktop\_Tools contains desktop administration and operating system tools, System\_Admin contains tools used by system administrators, and Information provides access to various help topics. Application Manager in the system includes these built-in application groups and the other applications and application groups to which the user has been granted access by the system.

Users can launch an application (and open an application window) either by double clicking on its application icon in Application Manager or pressing <Return> twice when focus is on the icon. In the

case of applications that use or create data files, users can launch the application when they double click on one of its file icons in File Manager or when they drop the file icon on the application icon; in this case, the data file is loaded into the application window that is opened. Finally, the Personal Applications subpanel in the Front Panel provides access to applications that are used frequently by users. If users have dragged an application icon from Application Manager to this subpanel, they can launch the application by clicking on its control in the subpanel or by dragging one of its file icons from File Manager to the control.

Users cannot rename or delete an application icon unless they have been granted permission to do so. Users can launch only one instance of an application from the desktop; double clicking on the icon for an application that is running only raises the application window to the front of the screen.

### **7.1.3.2 DII Configuration**

Built-in Applications. DII systems configure the set of utilities and tools available in the built-in application groups based on the privileges granted the user by the system. In particular, the Terminal Emulator application is not included in Application Manager for generic users, and the Terminal application is not available to them as a control in the Front Panel or from the File Menu in File Manager.

Installed Applications. Application Manager is the single repository for applications and application groups that the system adds to the desktop. DII applications are not installed in the Front Panel or its subpanels; only CDE components are installed there. However, if users choose to, they can drag frequently used applications to the Personal Applications subpanel and launch them from there. Only those applications which users have permission to execute are available in the Application Manager window; applications (as well as any files or folders used by the application) to which users do not have access are not visible anywhere on the desktop. Users can create new application groups within which to place the applications to which they are granted access.

Each application in the system is represented by a single application icon, with individual segments in the application available to users as defined in section 7.2. The label for the icon is the name of the application. Users can access all of the functionality in the application from this icon. A container defining an application group can be used instead of an application icon if other files (e.g., Read Me text, templates, sample data files) need to be bundled with the application. However, the application group includes an application icon that is used to launch the application. In addition, an application group may be used as the container for a suite of related applications (e.g., a Microsoft Office application group containing Word, Excel, and PowerPoint application icons).

## **7.1.4 File Management**

### **7.1.4.1 Accessing Files and Folders**

The system uses File Manager to display the files and folders available to users. Users can configure File Manager based on the way they want to see the content of folders (e.g., tree view, folder view); they can also change how objects are represented (e.g., by name, large icon, small icon, name, date, size) and the order in which icons are sorted (e.g., alphabetically, by file type, date, size). Opening a folder (e.g., by double clicking on it) changes the File Manager view to show the contents of that folder; opening a file starts the application that created the file and loads the contents of the file into the application window that is displayed.

Users can create new files and folders in the File Manager window. New folders use the standard icon provided by the desktop while new files use the icon defined by the application (see section 7.2.3.1 and appendix D). In both cases, user are required to name the object. Users can delete files and folders by dragging them to the Trash Can and then emptying the trash (i.e., choosing the Shred menu option

and then confirming the action); in addition, the Trash Can is emptied when users log out of a session. Deleting a file or folder moves the icon from its location on the desktop (e.g., the File Manager window) to the Trash Can window; emptying the trash removes the icon from this window and permanently deletes the object from the system.

#### **7.1.4.2 DII Configuration**

Terminal access. File Manager allows users to open a Terminal Emulator window with the same current folder as the File Manager window. As indicated previously, DII systems are to limit Terminal access to those users with this privilege.

### **7.1.5 Workspace Management**

#### **7.1.5.1 Using Workspaces**

The system uses Workspace Manager to control workspaces (e.g., the number available, which are open), workspace backdrops, and the Front Panel. Workspaces function as virtual desktops by increasing the amount of screen "real estate" available to users and creating customized areas where applications performing related functions can be placed. The system allows users to add, remove, and rename workspaces, to place applications in one, some, or all available workspaces, and to switch between workspaces.

A Workspace menu containing options for managing the workspace is available on all workspaces. Clicking BMenu on an empty part of the workspace displays the menu; access to this menu (i.e., to display the menu or choose a menu option) is not available from the keyboard. If any of the following workspace management functions are included in a Workspace menu, they are ordered: Shuffle Up, Shuffle Down, Refresh, Minimize/Restore Front Panel, Restart Workspace Manager, and Log Out. These options execute the actions listed in appendix C.

#### **7.1.5.2 DII Configuration**

TED enhancements. The system implements workspace enhancements provided by the TED version of CDE. Graphical Workspace Manager provides a miniature view of the contents of each workspace in the user's session and allows users to navigate among these workspaces, move applications between workspaces, and change the stacking order of application windows in each workspace. Users are able to access this manager from either a control on the Front Panel or from the Workspace menu. In addition, the Application List option is available in the Workspace menu so that users can move easily between applications running on the current or a different workspace. Finally, a Workspaces control is added to the set of user preference features available for configuration in Style Manager.

Multiple monitors. The desktop implementation defined here assumes a default hardware suite that includes a single monitor. If the system uses multiple monitors, it configures the desktop to display a separate Front Panel on each screen, with each screen functioning as a unique desktop environment.

### **7.1.6 Style Management**

#### **7.1.6.1 Customizing System Style**

Style Manager provides controls for desktop-wide customization of various features of system appearance and behavior. The system makes the following standard controls in Style Manager available to users:

Color - colors to be used in different areas of the screen (e.g., application windows, workspaces)

Font - font size to be used in window labels and text

Backdrop - pattern to be used to cover each workspace

Keyboard - volume of key clicks and rate at which characters repeat when users hold down their keys

Mouse - reversal of functions assigned to left and right buttons, assignment of the middle button to Adjust or Transfer functions, maximum time between clicks of a double click, and speed and threshold for accelerated pointer movement

Beep - volume, tone, and duration of auditory signals

Windows - focus policy, how active windows are displayed, and where window icons are displayed

Startup - session started at next login (e.g., current vs. home session), session to be the home session, and enable/disable logout confirmation

The system supports the Screen With Lock extension to the Screen control so that users can choose to enable or disable a screen saver and screen lock on timeout. In addition, the system implements a Workspaces control (see section 7.1.5.2) so that users can choose whether to synchronize workspaces on multiple screens (if they are using more than one monitor) and whether to hide or show workspace buttons.

### 7.1.6.2 DII Configuration

Color. Appendix D provides a group of color palettes that are appropriate for use in an office-like operational environment with normal ambient lighting; the Gray75 color palette is designated as the default. If the system is installed in this environment, it implements these color palettes (instead of the ones provided by CDE) in the Color control of Style Manager. If the system is installed in another environment where these palettes may be inappropriate, it implements the DII default and defines other system-specific palettes that are tailored to this environment. Any alternative color palettes are available to users in the Color control in Style Manager.

The system allows the desktop to choose the number of color sets used in a palette, with users able to select from among those available. Palettes are configured to generate foreground color dynamically based on the background color selected. The system implements the centralized color management capability provided by the desktop so that applications change color dynamically when users select a different palette. Applications that cannot use dynamic colors use the DII default palette as their color set (see appendix D).

DII systems should be able to constrain the color selection options available in the Color control of Style Manager. In particular, users should be limited to the predefined color palettes defined by the system. Users should be able to choose from the Number of Colors setting for each of these palettes but not allowed to modify individual colors in a palette or add or delete palettes. This latter capability is not currently available in CDE but is an enhancement desired in a future release of the product.

Font. The Font control in Style Manager provides users with a choice of seven font sizes in which to display text. The system implements the centralized font management capability provided by the desktop, using the default system and user fonts defined by the platform on which the system is installed (see appendix D). Applications use the fonts for which font aliases are defined in CDE (see appendix E). DII systems should be able to provide users with a choice of fonts as well as font sizes; this capability is an enhancement desired in a future release of CDE.

Windows. The Windows control allows users to select between implicit and explicit focus. DII systems should be able to configure this control to restrict users to a single focus policy or to allow them to choose between them; the preferred implementation for DII systems is to restrict users to explicit focus only.

## 7.2 APPLICATION DESIGN IN THE DII

Two models are available for the design of applications in DII systems, one based on centralized task management (CTM) within the application and the other based on distributed task management (DTM). In these models, an application may consist of a single segment, a parent and one or more nested segments, or multiple independent segments, or it may access a segment as a shared resource.

### 7.2.1 Applications Providing Centralized Task Management

A CTM application is designed to perform a single primary task or multiple related primary tasks (see figure 7-1). One task in the application is designated as the default; the primary window for that task opens when the application is launched and provides application control (i.e., P(AC) in figure 7-1). Navigation among tasks (i.e., access to other primary windows) is available only from within the application.

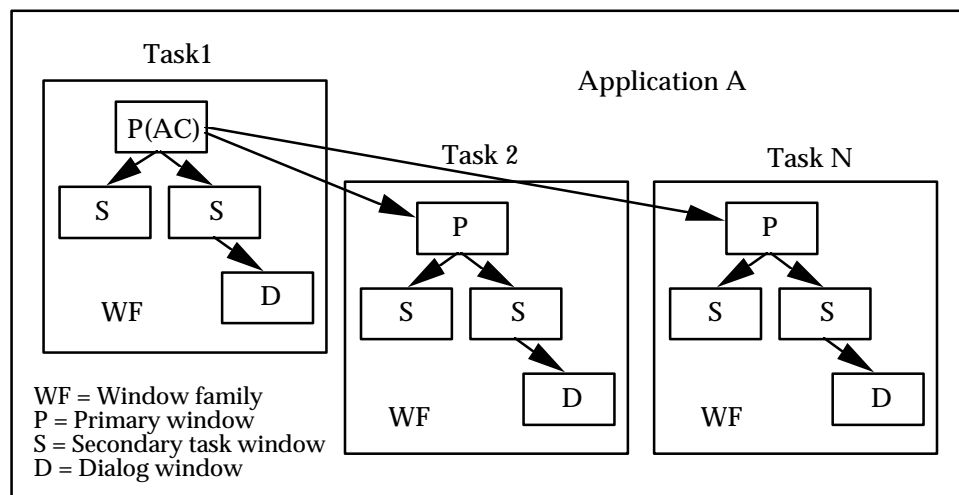


Figure 7-1. Design of a CTM application.

#### 7.2.1.1 Basic Implementation

**Launching.** Launching the application (using methods listed in section 7.1.3.1) opens the window with application control, as shown in figure 7-2. Navigation among tasks in the application may be limited to only this window or also be provided in the other primary windows in the application.

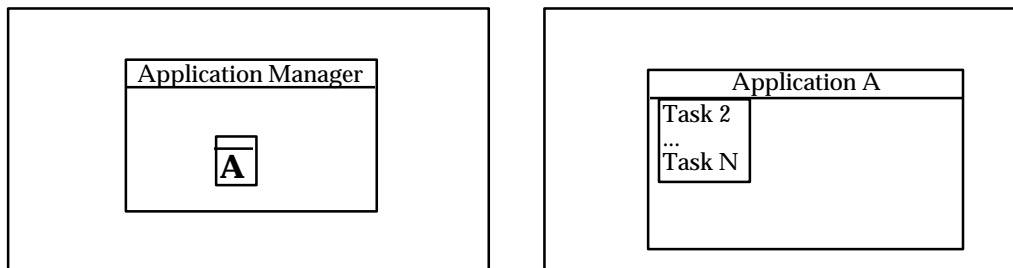


Figure 7-2. Launching a CTM application.

**Closing and exiting.** A Close option is available from the menu bar of each primary window in the application (including the one with application control); selecting this option closes all windows in the

family parented by the window and quits processing in the window.<sup>4</sup> Primary windows without a menu bar include a Close push button that performs this action. An Exit option is available only from the menu bar of the window with application control; selecting this option closes all windows and quits processing in the application.

**Window naming.** The title of the window with application control is the name of the application; other primary windows are identified by application and task name, with no separator between the names, as shown in figure 7-3. Primary windows in which a file has been loaded are identified by application name, followed by a hyphen, and then the name of the file; the hyphen is preceded and followed by spaces. Secondary windows are identified by application name, followed by a colon, and then the name of the secondary window; the colon is preceded and followed by spaces. The name of the application can be abbreviated if space is limited.

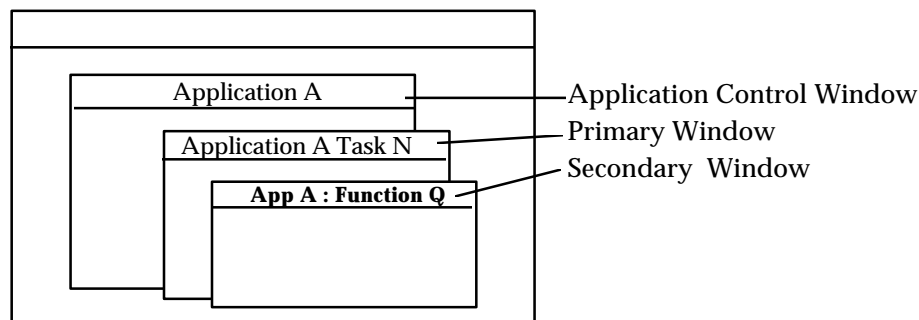


Figure 7-3. Window naming conventions in CTM applications.

### 7.2.1.2 Nested Implementation

The basic implementation of a CTM application assumes that the software for each task comes from a single source (i.e., one segment). Using different segments to perform the tasks in the application results in a nested implementation in which one segment is launched from within another (see figure 7-4). In this implementation, users would view the child segment as one of the tasks available within the parent segment.

<sup>4</sup> Per version 1.2 of the Motif Style Guide: "When an application has multiple primary windows, the window manager should treat each primary window as if it were an independent application" (p 7-4). As a result, selecting the Close option in the application control window closes only the windows in the family parented by this window and not all of the windows in the application.

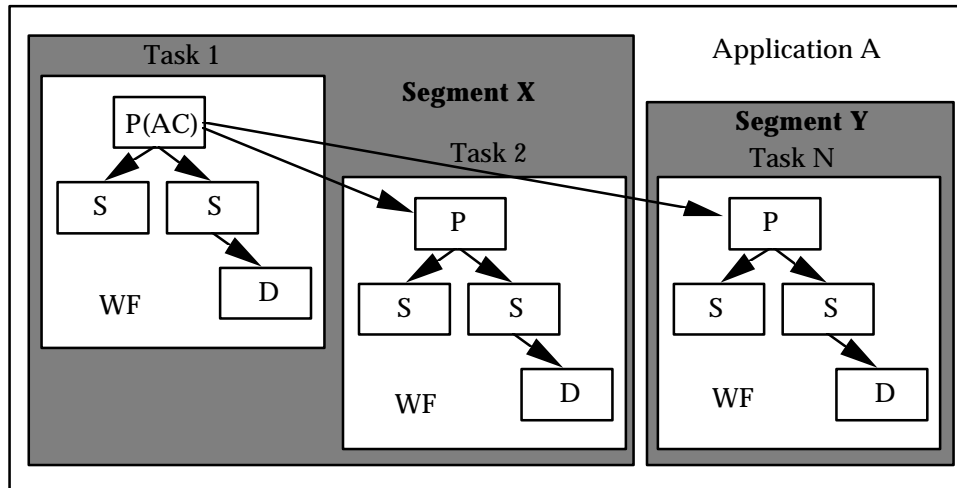


Figure 7-4. Nested segments within a CTM application.

In a nested implementation, the parent segment provides application control for the overall application. The application icon is named for the parent segment; double clicking on the icon launches this segment. Closing a primary window in the parent segment has no effect on windows in a child segment. Exiting the window with application control in the parent segment closes all of the windows and quits processing in both parent and child segments.

The child segment is not available on the desktop and can only be launched from within the parent segment. A Close option is available from the menu bar of the primary window(s) in the child segment; selecting this option closes all windows in the family parented by the window and quits processing in the window. An Exit option can be included in the primary window of the child segment, or it can rely on the parent segment to provide this option. The name of the parent segment is used as the application name in the title of all primary windows in the child segment.

### 7.2.2 Applications Providing Distributed Task Management

A DTM application consists of multiple top-level tasks that can be opened and exited independently (see figure 7-5). Navigation among tasks is available from both the application icon on the desktop and within the application. One of the tasks is designated as the default for the application; the primary window for this task opens when the application is launched from the desktop. Application control is not assigned to the default window; instead, it is distributed among the primary windows in the application.<sup>5</sup>

<sup>5</sup> This model differs from the approach defined in the Bellcore style guide for applications consisting of multiple primary tasks. In that document, each top-level task (i.e., window family) is represented by an icon on the desktop, with application control assigned to a single primary window in the application.



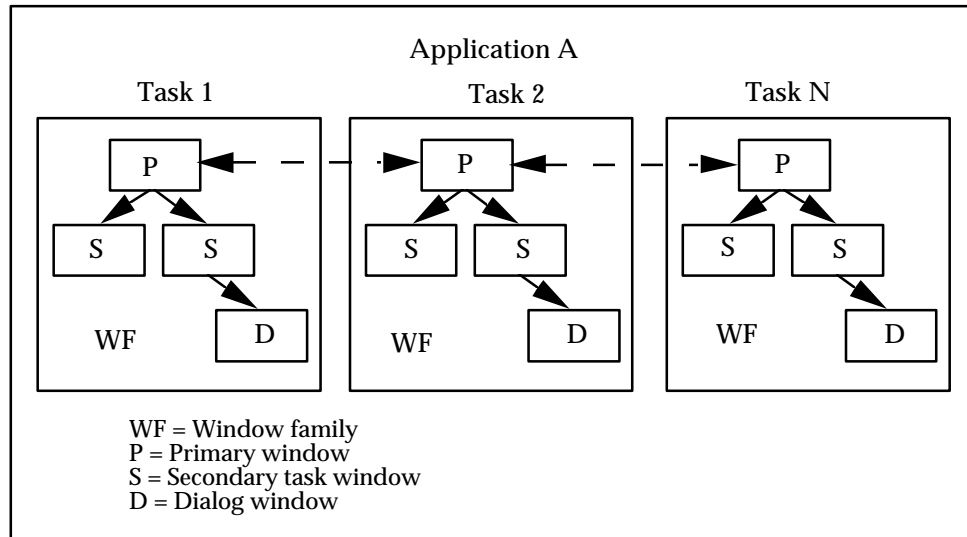


Figure 7-5. Design of a DTM application.

The tasks in a DTM application may be performed by software from a single source or from multiple sources. For example, a logistics application might support tasks related to data query, message generation, and data display, with the software for each task contributed by either the same or different segments. Because a DTM application consists of multiple independent tasks, the basic implementation defined here applies to both single-source and multi-source applications.

#### 7.2.2.1 Basic Implementation

**Launching.** A pop-up menu is attached to the application icon listing all of the top-level tasks in the application (as shown in the upper part of figure 7-6). Double clicking on the icon launches the default task and open its primary window. Selecting one of the tasks from the pop-up menu launches the task and opens its primary window. Launching any of the tasks in the application changes the pop-up menu option for the task to unavailable. In addition, each primary window includes menu options from which all of the other top-level tasks in the application can be accessed. These menu options can be included in a single menu or distributed among several menus (as shown in the lower part of figure 7-6).

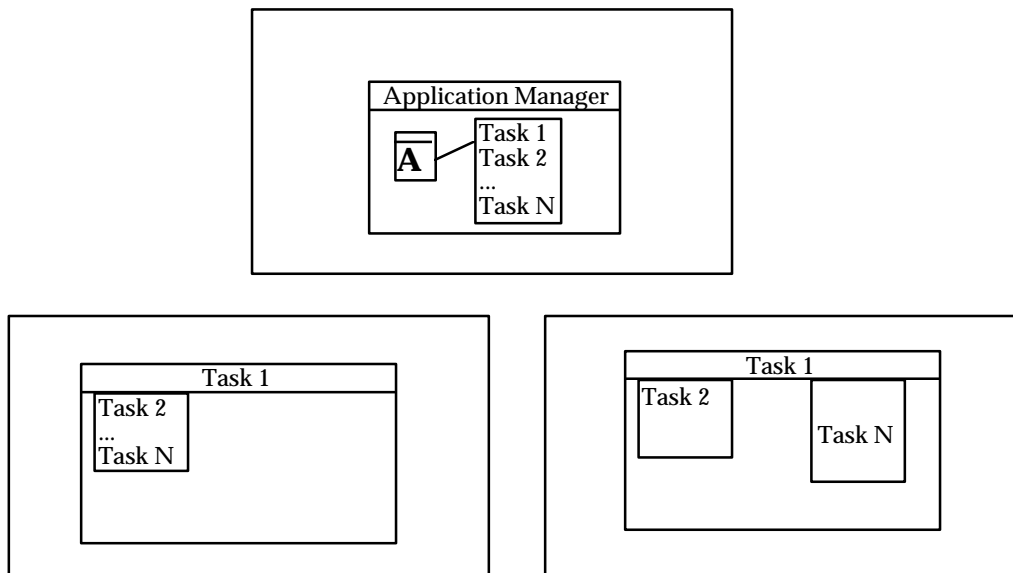


Figure 7-6. Launching a DTM application.

**Closing and exiting.** Close and Exit options are available from the menu bar of each primary window in the application. Selecting either of these options performs the same action, i.e., closes the windows in the family parented by the window and quits processing in the window (i.e., the task). Exiting the last primary window in the application also exits the application. An Exit All option can be included in each primary window so that users can exit each task separately or exit the entire application in a single action. When this option is selected, each primary window receives focus prior to being exited, with a prompt displayed if the window contains unsaved data. The last primary window exited also exits the application.

**Window naming.** Primary windows in the application are identified by task name, and secondary windows are identified by task name, followed by a colon, and then the name of the secondary window; the colon is preceded and followed by spaces (see figure 7-7).<sup>6</sup>

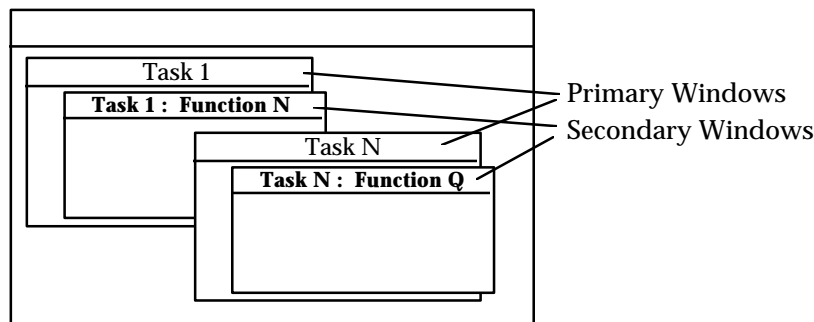


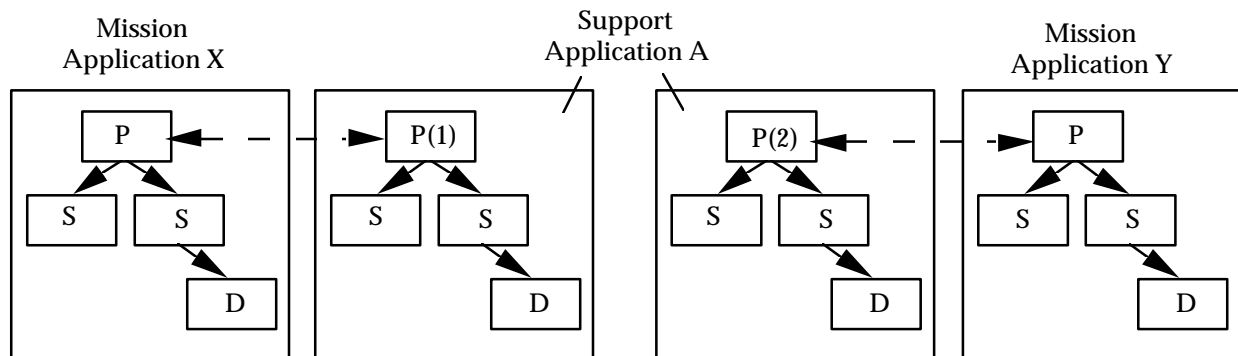
Figure 7-7. Window naming conventions in DTM applications.

### 7.2.2.2 Resource Sharing Among Applications

The DTM application model can be extended to support the implementation of resource sharing among applications. In this case, common support applications within a system provide generic services which users are able to access independently; these applications also make their services available to mission applications, either individually or as a shared resource. For example, Support Application A might provide basic chart functionality that can be accessed from the desktop. This functionality would also be available as a “task” within Mission Applications X and Y. Each application could open separate map windows (i.e., P(1) and P(2) in figure 7-8a) for plotting their data, or they could share a single map window in order to provide a common tactical picture (figure 7-8b).

<sup>6</sup> Primary windows for each task are not linked to the application by name because the tasks in a given application can be performed by different segments, each of which may be available in more than one application. One disadvantage of not including application name is that users cannot easily identify the set of tasks that will be exited if they select an Exit All option in one of the windows.

(a) If Application A is accessed independently by Applications X and Y:



(b) If Application A is accessed as a shared resource by Applications X and Y:

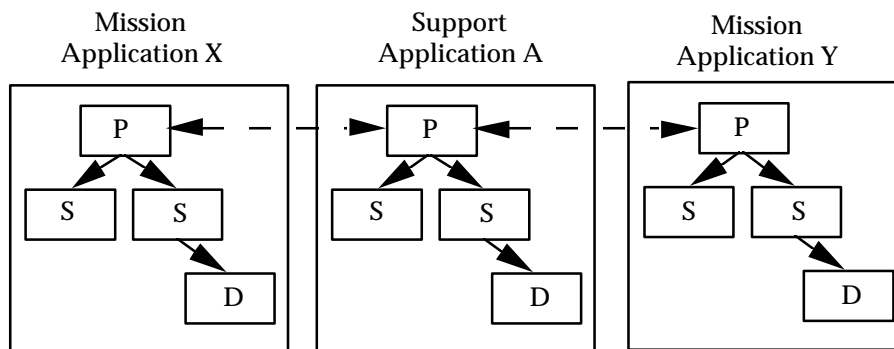
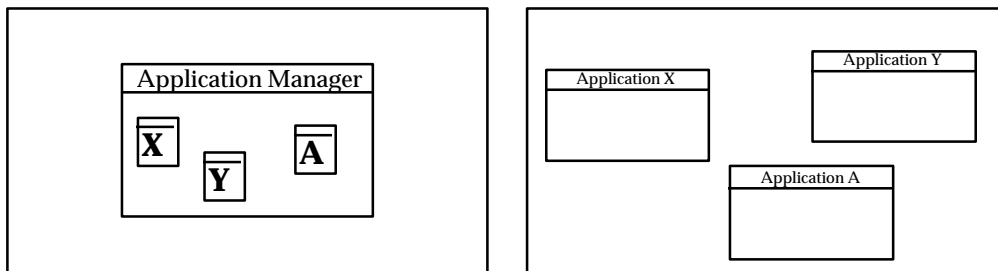


Figure 7-8. Providing independent or shared access to support services.

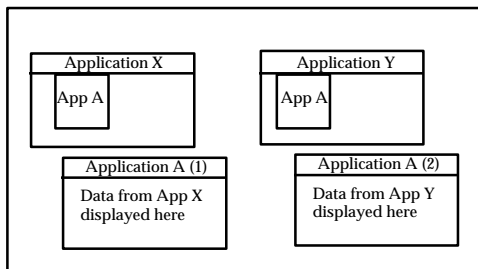
Launching Application A, X, or Y from the desktop opens the primary window for the application (see figure 7-9a). Launching Application A from a menu option within either Application X or Y changes the appearance of the option to unavailable; when separate windows from Application A are opened, each instance is identified by number (see figure 7-9b).<sup>7</sup>

<sup>7</sup> A weakness of this approach is that support application windows are not linked to the associated mission application by name (e.g., so that users can easily distinguish the map window displaying data from App X from the one displaying data from App Y).

(a) If Application A is launched as an independent application (similar to Applications X and Y):



(b) If Application A is launched individually from within Application X or Y:



(c) If Application A is launched as a shared resource by Application X or Y:

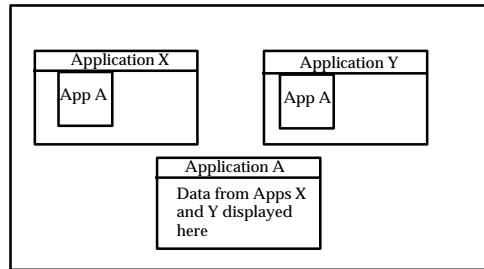


Figure 7-9. Options for launching a support application in a shared-resource implementation.

Each mission and support application is closed and exited independently; executing these actions in any of the applications affects only the windows in the application. When Application A is launched as a shared resource (see figure 7-9c) and then Application X or Y is exited, the shared window remains open but the data from the application being exited are removed from the window. When Application A is launched from within another application but is not shared, the latter application can provide an Exit All option to exit both it and Application A. When Application A is a shared resource, Exit All is not be available (since the shared window may contain data from an application other than the one being exited).

## 7.2.3 Other Design Considerations

### 7.2.3.1 Icon Design in CDE (Motif Only)

A key feature of CDE is the use of icons to represent both an application and the files it creates. Appendix D describes CDE requirements with respect to icon design. The applications provides the icon set in the three sizes and two formats indicated in this appendix.<sup>8</sup> All of the icons in the set are designed to a common theme (e.g., a similar graphic image), with individual icons tailored to reflect specific application functions. The application icon has a three-dimensional style and can fill the entire icon bounding box or be irregular in shape. The container icon (e.g., an application group) is designed so that its shape provides cues as to its function and the kind of objects it contains. Document and file icons indicate what application is associated with the document or file and the kind of data stored in the icon; if an application supports multiple file formats, document icons use the same basic file graphic but have a different "tag" to distinguish each output format. Finally, if the application may be used by non-English audiences, icons are designed to use international symbols whenever possible (see section 12 on internationalization).

<sup>8</sup> Additional direction with regard to icon design, including a standard set of icon graphics, will be provided in a future version of this document.

### **7.2.3.2 Accessing Segments Within an Application**

In applications composed of multiple segments, the tasks in each segment can occupy all or part of an application menu, be available in a submenu, or be distributed across several menus. If menus become excessively long (and extend beyond the bottom of the screen), the application uses submenus, following guidelines in section 5.5, to reduce menu length. Menus that cannot be shortened include controls (e.g., arrow buttons for scrolling) for users to view and select the options that extend beyond the bottom of the screen. The application determines when users can and cannot access each of the tasks performed by a segment and dims the associated menu option to indicate its unavailability.